

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Claims 1-37. (cancelled)

38. (currently amended) A method for performing a plurality of virtualization services, said method being further operative to perform said virtualization services within a data path, ~~said method comprising comprises the steps of:~~

~~[[a]]-initiating an internet small computer storage interface (iSCSI) with a least an initiator host;~~

~~receiving a logic command from said initiator host over an internet protocol (IP) network to be performed on at least one virtual volume, said logic command including at least a virtual address of at least one virtual volume;~~

~~[[d]]-scheduling said logic command for execution, wherein said logic command is at least a SCSI command comprising the steps of:~~

~~1) initiating an iSCSI session with an initiator host including the step of:~~

~~— a) determining if said initiator host is authorized to send execute said logic command; and,~~

~~b) denying said logic command from said initiator host, if said initiator host is unauthorized;~~

~~— 2) receiving said logic command from said initiator host;~~

~~3) parsing said logic command to determine at least said virtual address and said logic command's type;~~

4) performing a check to determine if said logic command is valid;

5) generating a response command if said logic command is invalid,

-otherwise, adding said logic command to a host-logical unit (LU) queue;

scheduling said logic command for execution; and,

6) generating a data transfer request.

[[e]] translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device,

wherein the translation is performed using a mapping information including at least the relations between said at least one virtual volume and its respective logical units (LUs), and storage devices;

[[d]] determining, using a check point list, the amount of data to be transferred via a IP network, wherein the check-point list includes a linked list of data chunks; and,

[[e]] executing said physical commands on said storage devices.

39. (previously presented) The method of claim 38, wherein said response command comprises an iSCSI service response code indicating the type of a generated error.

40. (previously presented) The method of claim 38, wherein said host-LU queue comprises logic commands requested to be executed by said host on said LU.

41. (previously presented) The method of claim 38, wherein scheduling said logic

command for execution further comprises the step of: selecting said logic command to be executed from said host-LU queue.

42. (original) The method of claim 41, wherein the selection is performed using at least one of the following selection algorithms: recently used, round robin, weighted round robin, random, least loaded LU.

43. (currently amended) The method of claim 38, wherein said command type is a read SCSI command.

44. (original) The method of claim 43, wherein said amount of data to be transferred is determined by an available space parameter.

45. (original) The method of claim 44, wherein said available space parameter defines the number of data bytes to be sent to the host.

46. (currently amended) The method of claim 44, wherein the following steps comprise executing said physical commands on said storage devices, further comprising:

- a) accessing a storage device using a physical address;
- b) retrieving from said accessed storage device the number of bytes designated in said available space parameter;
- c) sending the retrieved data to said host; and,

d) repeating said steps a) through d) until all requested data is read from said storage devices.

47. (original) The method of claim 46, wherein said physical commands are executed in parallel.

48. (currently amended) The method of claim 38, wherein said command type is a write SCSI command.

Claim 49. (cancelled)

50. (currently amended) A method for performing a plurality of virtualization services, said method being further operative to perform said virtualization services within a data path, said method comprises comprising the steps of:

a) receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address;

b) scheduling said logic command for execution;

c) translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device, wherein the translation is performed using a mapping information including at least the relations between said at least one virtual volume and its respective logical units (LUs) and storage devices;

d)-determining, using a check-point list, the amount of data to be transferred ~~via~~
 across a an internet protocol (IP) network, wherein said check-point list further defines
 how data should be sent from an initiator host to said storage device; and,
 e)-executing said physical commands on said storage devices.

51. (original) The method of claim 50, wherein said check-point list comprises a linked
list of data chunks.

52. (currently amended) The method of claim 51, wherein ~~the following steps comprise~~
executing said physical commands on said storage devices, further comprising:
 a) filling at least one data chunk with said data retrieved from ~~the~~ said IP network;
 b) accessing said storage device using a physical address in the translated
physical command;
 c) writing said data chunk to said accessed storage device; and,
 d) repeating said steps a) through d) for all data chunks in said check-point list.

53. (original) The method of claim 52, wherein said physical commands are executed in
parallel.

54. (previously presented) The method of claim 53, wherein said physical commands
are constructed in a data structure.

55. (original)The method of claim 54, wherein said data structure further includes at least one of: an alternative command link, a pointer to said storage device.

56. (original)The method of claim 55, wherein said alternative command link links between at least two physical commands that can be executed in parallel.

Claims 57 - 67. (cancelled)

68. (currently amended) A computer executable code for performing a plurality of virtualization services on a recordable media, said computer executable code being further operative to perform said virtualization services within a data path, comprising said code comprises the steps of:

initiating an internet small computer storage interface (iSCSI) with a least an initiator host;

[(a)] receiving a logic command from said initiator host over an internet protocol (IP) network to be performed on at least one virtual volume, said logic command including at least a virtual address of at least one virtual volume;

[(d)] scheduling said logic command for execution, wherein said logic command is at least a SCSI command comprising the steps of:

1) initiating an iSCSI session with an initiator host including the step of:

— a) determining if said initiator host is authorized to send execute said logic command;

and,

b) denying said logic command from said initiator host, if said initiator host is unauthorized;

— 2) receiving said logic command from said initiator host;

3) parsing said logic command to determine at least said virtual address and said logic command's type;

4) performing a check to determine if said logic command is valid;

5) generating a response command if said logic command is invalid, otherwise, adding said logic command to a host-logical unit (LU) queue; and,

scheduling said logic command for execution;

6) generating a data transfer request.

[[c]] translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device, wherein the translation is performed using a mapping information including at least the relations between said at least one virtual volume and its respective logical units (LUs), and storage devices;

[[d]] determining, using a check point list, the amount of data to be transferred across said IP via a network, wherein the check-point list includes a linked list of data chunks; and,

[[e]] executing said physical commands on said storage devices.

Claims 69 - 79. (cancelled)

80. (currently amended) A computer product stored on a computer-readable medium comprising software instructions operable to enable a computer to perform a process for

performing a plurality of virtualization services, said process being further operative to perform said virtualization services within a data path, comprising ~~said code comprises the steps of:~~

- a)-receiving a logic command to be performed on at least one virtual volume, said logic command including at least a virtual address;
- b)-scheduling said logic command for execution;
- c)-translating, in one pass, said logic command to a list of physical commands, wherein each of said physical commands is targeted to a different storage device, wherein the translation is performed using a mapping information including at least the relations between said at least one virtual volume and its respective logical units (LUs) and storage devices;
- d)-determining, using a check-point list, the amount of data to be transferred across a an internet protocol (IP) network, wherein said check-point list further defines how data should be sent from an initiator host to said storage device; and,
- e)-executing said physical commands on said storage devices.

81. (previously presented) The computer program product of claim 80, wherein said check-point list comprises a linked list of data chunks.

82. (currently amended) The computer program product of claim 81, wherein the following steps comprise executing said physical commands on said storage devices, further comprising:

- a) filling at least one data chunk with said data retrieved from the said IP network;
- b) accessing said storage device using a physical address in the translated physical command;
- c) writing said data chunk to said accessed storage device; and,
- d) repeating said steps a) through d) for all data chunks in said check-point list.

83. (previously presented) The computer program product of claim 82, wherein said physical commands are executed in parallel.

84. (previously presented) The computer program product of claim 83, wherein said physical commands are constructed in a data structure.

85. (previously presented) The computer program product of claim 84, wherein said data structure further includes at least one of: an alternative command link, a pointer to said storage device.

86. (previously presented) The computer program product of claim 85, wherein said alternative command link links between at least two physical commands that can be executed in parallel.

Claims 87 - 97. (cancelled)